```
FFFFFFFFFFFFFFF        111            111       XXX        XXX
FFFFFFFFFFFFFFF        111            111       XXX        XXX
FFFFFFFFFFFFFFF        111            111       XXX        XXX
FFF                111111         111111       XXX        XXX
FFF                111111         111111       XXX        XXX
FFF                111111         111111       XXX        XXX
FFF                    111            111          XXX   XXX
FFF                    111            111          XXX   XXX
FFF                    111            111          XXX   XXX
FFFFFFFF.FFF           111            111             XXX
FFFFFFFFFFFF           111            111             XXX
FFFFFFFFFFFF           111            111             XXX
FFF                    111            111          XXX   XXX
FFF                    111            111          XXX   XXX
FFF                    111            111          XXX   XXX
FFF                    111            111       XXX        XXX
FFF                    111            111       XXX        XXX
FFF                    111            111       XXX        XXX
FFF                    111            111       XXX        XXX
FFF                    111            111       XXX        XXX
FFF                111111111      111111111     XXX        XXX
FFF                111111111      111111111     XXX        XXX
FFF                111111111      111111111     XXX        XXX
```

```
_$29

Symb
-----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

FILEID**EXTIDX

```
EEEEEEEEEE  XX      XX  TTTTTTTTTT    IIIIII    DDDDDDDD    XX      XX
EEEEEEEEEE  XX      XX  TTTTTTTTTT    IIIIII    DDDDDDDD    XX      XX
EE          XX      XX      TT          II      DD      DD  XX      XX
EE          XX      XX      TT          II      DD      DD  XX      XX
EE            XX  XX        TT          II      DD      DD    XX  XX
EE            XX  XX        TT          II      DD      DD    XX  XX
EEEEEEEE        XX          TT          II      DD      DD      XX
EEEEEEEE        XX          TT          II      DD      DD      XX
EE            XX  XX        TT          II      DD      DD    XX  XX
EE            XX  XX        TT          II      DD      DD    XX  XX
EE          XX      XX      TT          II      DD      DD  XX      XX      ....
EE          XX      XX      TT          II      DD      DD  XX      XX      ....
EEEEEEEEEE  XX      XX      TT        IIIIII    DDDDDDDD    XX      XX      ....
EEEEEEEEEE  XX      XX      TT        IIIIII    DDDDDDDD    XX      XX      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE EXTIDX (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This routine extends the volume's index file.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  14-Apr-1977  10:44
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-006 ACG0438         Andrew C. Goldstein,    2-Aug-1984  11:54
1 !               Add check of index file bitmap size to max files check
1 !
1 !       V03-005 CDS0003         Christian D. Saether    25-Apr-1984
1 !               Long addressing mode for certain external routines.
1 !
1 !       V03-004 ACG0409         Andrew C. Goldstein,    21-Mar-1984  21:58
```

```
0058  1 !                Remove file number parameter; set index file EOF to end.
0059  1 !
0060  1 !        V03-003 CDS0002         Christian D. Saether    30-Dec-1983
0061  1 !                Use L_NORM linkage and BIND_COMMON macro.
0062  1 !
0063  1 !        V03-002 CDS0001         Christian D. Saether    20-Sep-1983
0064  1 !                Serialize reading of index file header.
0065  1 !                Stuff FH2$L_HIGHWATER if appropriate.
0066  1 !
0067  1 !        V03-001 ACG0274         Andrew C. Goldstein,    23-Mar-1982  11:25
0068  1 !                Allow use of alternate index file header
0069  1 !
0070  1 !        V02-002 STJ34965        Steven T. Jeffreys,     28-Feb-1981
0071  1 !                Temporary fix to set FIB$V_NOCHARGE bit for the
0072  1 !                index file, so that if will not be charged for diskquota.
0073  1 !
0074  1 !        B0101   ACG0121         Andrew C. Goldstein,    16-Jan-1980  21:39
0075  1 !                Make context save and restore into subroutines
0076  1 !
0077  1 !        B0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:00
0078  1 !        Previous revision history moved to [F11B.SRC]F11B.REV
0079  1 !**
0080  1
0081  1
0082  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
0083  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;  85          1074  1 GLOBAL ROUTINE EXTEND_INDEX : L_NORM NOVALUE =
;  86          1075  1
;  87          1076  1 !++
;  88          1077  1 !
;  89          1078  1 ! FUNCTIONAL DESCRIPTION:
;  90          1079  1 !
;  91          1080  1 !       This routine extends the volume's index file.
;  92          1081  1 !
;  93          1082  1 ! CALLING SEQUENCE:
;  94          1083  1 !       EXTEND_INDEX ()
;  95          1084  1 !
;  96          1085  1 ! INPUT PARAMETERS:
;  97          1086  1 !       NONE
;  98          1087  1 !
;  99          1088  1 ! IMPLICIT INPUTS:
; 100          1089  1 !       CURRENT_VCB: address of volume VCB
; 101          1090  1 !
; 102          1091  1 ! OUTPUT PARAMETERS:
; 103          1092  1 !       NONE
; 104          1093  1 !
; 105          1094  1 ! IMPLICIT OUTPUTS:
; 106          1095  1 !       NONE
; 107          1096  1 !
; 108          1097  1 ! ROUTINE VALUE:
; 109          1098  1 !       NONE
; 110          1099  1 !
; 111          1100  1 ! SIDE EFFECTS:
; 112          1101  1 !       index file extended, index file window and index file FCB modified
; 113          1102  1 !
; 114          1103  1 !--
; 115          1104  1
; 116          1105  2 BEGIN
; 117          1106  2
; 118          1107  2 LOCAL
; 119          1108  2         MAX_FILES,                      ! max number of files on volume
; 120          1109  2         FILESIZE,                       ! size of index file after extend.
; 121          1110  2         FIB             : REF BBLOCK,   ! address of FIB for extend operation
; 122          1111  2         HEADER          : REF BBLOCK,   ! address of index file header
; 123          1112  2         FCB             : REF BBLOCK,   ! address of index file FCB
; 124          1113  2         WINDOW          : REF BBLOCK,   ! address of index file window
; 125          1114  2         FREE_POINTERS,                  ! number of free retrieval pointers
; 126          1115  2                                         ! in index file window
; 127          1116  2         FILES_TO_GO,                    ! number of files likely to be created
; 128          1117  2                                         ! on this volume
; 129          1118  2         BLOCKS_NEEDED;                  ! amount to extend index file by
; 130          1119  2
; 131          1120  2 BIND_COMMON;
; 132          1121  2
; 133          1122  2 EXTERNAL ROUTINE
; 134          1123  2         SERIAL_FILE     : L_NORM,
; 135          1124  2         RELEASE_SERIAL_LOCK     : L_NORM,
; 136          1125  2         SAVE_CONTEXT    : L_NORM,       ! save reentrant context area
; 137          1126  2         RESTORE_CONTEXT : L_NORM,       ! restore reentrant context area
; 138          1127  2         READ_IDX_HEADER : L_NORM,       ! read index file header
; 139          1128  2         TURN_WINDOW     : L_NORM ADDRESSING_MODE (GENERAL), ! update file window
; 140          1129  2         EXTEND          : L_NORM,       ! extend a file
; 141          1130  2         CHECKSUM        : L_NORM,       ! compute file header checksum
```

```
;  142          1131  2          WRITE_HEADER     : L_NORM,        ! write back file header
;  143          1132  2          RESET_LBN        : L_NORM,        ! reassign LBN of buffer
;  144          1133  2          WRITE_BLOCK      : L_NORM,        ! write a disk block
;  145          1134  2          INVALIDATE       : L_NORM,        ! invalidate buffer
;  146          1135  2          INIT_FCB2        : L_NORM;        ! update file control block
;  147          1136  2
;  148          1137  2
;  149          1138  2 ! Extending the index file is a secondary operation, so we must save away the
;  150          1139  2 ! primary context, and then set up the appropriate context for this operation.
;  151          1140  2 !
;  152          1141  2
;  153          1142  2 SAVE_CONTEXT ();
;  154          1143  2 FIB = SECOND_FIB;
;  155          1144  2 CH$FILL (0, FIB$C_LENGTH, .FIB);
;  156          1145  2 FIB[FIB$W_FID_NUM] = 1;
;  157          1146  2 FIB[FIB$W_FID_SEQ] = 1;
;  158          1147  2
;  159          1148  2 PRIMARY_FCB = FCB = .CURRENT_VCB[VCB$L_FCBFL];
;  160          1149  2 CURRENT_WINDOW = WINDOW = .FCB[FCB$L_WCFL];
;  161          1150  2
;  162          1151  2 ! Serialize on the index file header.
;  163          1152  2 !
;  164          1153  2
;  165          1154  2 PRIM_LCKINDX = SERIAL_FILE (FIB [FIB$W_FID]);
;  166          1155  2
;  167          1156  2 ! Read the index file header.
;  168          1157  2 !
;  169          1158  2
;  170          1159  2 HEADER = READ_IDX_HEADER ();
;  171          1160  2
;  172          1161  2 ! Turn the index file window to VBN 3. Then compute the number of free
;  173          1162  2 ! retrieval pointers in the index file window, discounting pointers (if
;  174          1163  2 ! any) that only map the boot and home block.
;  175          1164  2 !
;  176          1165  2
;  177          1166  2 KERNEL_CALL (TURN_WINDOW, .WINDOW, .HEADER, 3, 1);
;  178          1167  2
;  179          1168  2 FREE_POINTERS = (.WINDOW[WCB$W_SIZE]-WCB$C_LENGTH)/6 - .WINDOW[WCB$W_NMAP];
;  180          1169  2 IF .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT] LEQU 3
;  181          1170  2 THEN
;  182          1171  3     BEGIN
;  183          1172  3     FREE_POINTERS = .FREE_POINTERS + 1;
;  184          1173  3     IF .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT] + .WINDOW[WCB$W_P2_COUNT] LEQU 3
;  185          1174  3     THEN FREE_POINTERS = .FREE_POINTERS + 1;
;  186          1175  2     END;
;  187          1176  2 IF .FREE_POINTERS LEQ 0 THEN FREE_POINTERS = 1;
;  188          1177  2
;  189          1178  2 ! Check the amount of index file (as determined by max files and the
;  190          1179  2 ! size of the index file bitmap).
;  191          1180  2 !
;  192          1181  2
;  193          1182  2 MAX_FILES = MINU (.CURRENT_VCB[VCB$L_MAXFILES], .CURRENT_VCB[VCB$B_IBMAPSIZE] * 4096);
;  194          1183  2 FILES_TO_GO = .MAX_FILES - .FCB[FCB$L_FILESIZE]
;  195          1184  2             + .CURRENT_VCB[VCB$B_IBMAPSIZE] + .CURRENT_VCB[VCB$W_CLUSTER]*4;
;  196          1185  2 IF .FILES_TO_GO LEQ 0
;  197          1186  2 THEN ERR_EXIT (SS$_IDXFILEFULL);
;  198          1187  2
```

```
;  199          1188  2 ! Compute the number of files likely to still be created on the volume.  This
;  200          1189  2 ! is the minimum of the number permitted minus the current number and a
;  201          1190  2 ! fraction of the number of free blocks on the volume. The amount to extend
;  202          1191  2 ! the index file by is this quantity divided by the number of available
;  203          1192  2 ! retrieval pointers in the index file window.
;  204          1193  2 !
;  205          1194  2
;  206          1195  2 FILES_TO_GO = MINU (.FILES_TO_GO,
;  207          1196  2                     .CURRENT_VCB[VCB$L_FREE] / .CURRENT_VCB[VCB$W_CLUSTER] / 4);
;  208          1197  2
;  209          1198  2 BLOCKS_NEEDED = MINU (.FILES_TO_GO / .FREE_POINTERS, 1000);
;  210          1199  2
;  211          1200  2 ! Build the extend control in the FIB and call the EXTEND routine.
;  212          1201  2 !
;  213          1202  2
;  214          1203  2 FIB[FIB$L_EXSZ] = .BLOCKS_NEEDED;
;  215          1204  2 FIB[FIB$V_ALCON] = 1;
;  216          1205  2 FIB[FIB$V_ALCONB] = 1;
;  217          1206  2 FIB[FIB$V_ALDEF] = 1;
;  218          1207  2 FIB[FIB$V_NOHDREXT] = 1;
;  219          1208  2
;  220          1209  2 ! Set NOCHARGE bit so the index file will not be charged for diskquota.
;  221          1210  2 !
;  222          1211  2 FIB [FIB$V_NOCHARGE] = 1;
;  223          1212  2
;  224          1213  2 EXTEND (.FIB, .HEADER);
;  225          1214  2
;  226          1215  2 FILESIZE = .FIB[FIB$L_EXSZ] + .FIB[FIB$L_EXVBN] - 1;
;  227          1216  2 BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBK] = ROT (.FILESIZE, 16);
;  228          1217  2 IF NOT .CURRENT_VCB[VCB$V_NOHIGHWATER]
;  229          1218  2 THEN BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (.FILESIZE+1, 16);
;  230          1219  2
;  231          1220  2 ! If this file header supports it, stuff the high water field to
;  232          1221  2 ! be the allocated size.
;  233          1222  2 !
;  234          1223  2
;  235          1224  2 IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
;  236          1225  2 THEN
;  237          1226  2     HEADER [FH2$L_HIGHWATER] = .FILESIZE + 1;
;  238          1227  2
;  239          1228  2 ! Now write the header, update the FCB, and restore the primary context.
;  240          1229  2 !
;  241          1230  2
;  242          1231  2 CHECKSUM (.HEADER);
;  243          1232  2 WRITE_HEADER ();
;  244          1233  2 RESET_LBN (.HEADER, .CURRENT_VCB[VCB$L_IXHDR2LBN]);
;  245          1234  2 WRITE_BLOCK (.HEADER);
;  246          1235  2 INVALIDATE (.HEADER);
;  247          1236  2 KERNEL_CALL (INIT_FCB2, .FCB, .HEADER);
;  248          1237  2
;  249          1238  2 RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
;  250          1239  2 PRIM_LCKINDX = 0;
;  251          1240  2
;  252          1241  2 RESTORE_CONTEXT ();
;  253          1242  2 USER_STATUS[1] = 0;
;  254          1243  2
;  255          1244  1 END;                                    ! end of routine EXTEND_INDEX
```

```
                                                        .TITLE  EXTIDX
                                                        .IDENT  \V04-000\

                                                        .EXTRN  SERIAL_FILE, RELEASE_SERIAL_LOCK
                                                        .EXTRN  SAVE_CONTEXT, RESTORE_CONTEXT
                                                        .EXTRN  READ_IDX_HEADER
                                                        .EXTRN  TURN_WINDOW, EXTEND
                                                        .EXTRN  CHECKSUM, WRITE_HEADER
                                                        .EXTRN  RESET_LBN, WRITE_BLOCK
                                                        .EXTRN  INVALIDATE, INIT_FCB2

                                                        .PSECT  $CODE$,NOWRT,2

                                    01FC 00000          .ENTRY  EXTEND_INDEX, Save R2,R3,R4,R5,R6,R7,R8     ; 1074
                     58        80   AA  9E 00002        MOVAB   -128(BASE), R8                              ; 1118
                     57        98   AA  9E 00006        MOVAB   -104(BASE), R7                              ;
            0000G    CF             00  FB 0000A        CALLS   #0, SAVE_CONTEXT                            ; 1142
                     56      0244   CA  9E 0000F        MOVAB   580(BASE), FIB                              ; 1143
0040   8F      00    6E             00  2C 00014        MOVC5   #0, (SP), #0, #64, (FIB)                    ; 1144
                                    66     0001B                                                            ;
               04    A6 00010001    8F  D0 0001C        MOVL    #65537, 4(FIB)                              ; 1145
                     54        00   B7  D0 00024        MOVL    @0(R7), FCB                                 ; 1148
               08    AA             54  D0 00028        MOVL    FCB, 8(BASE)                                ;
                     52        10   A4  D0 0002C        MOVL    16(FCB), WINDOW                             ; 1149
               0C    AA             52  D0 00030        MOVL    WINDOW, 12(BASE)                            ;
                               04   A6  9F 00034        PUSHAB  4(FIB)                                      ; 1154
            0000G    CF             01  FB 00037        CALLS   #1, SERIAL_FILE                             ;
               18    AA             50  D0 0003C        MOVL    R0, 24(BASE)                                ;
            0000G    CF             00  FB 00040        CALLS   #0, READ_IDX_HEADER                         ; 1159
                     53             50  D0 00045        MOVL    R0, HEADER                                  ;
                                    01  DD 00048        PUSHL   #1                                          ; 1166
                                    03  DD 0004A        PUSHL   #3                                          ;
                                    0C  BB 0004C        PUSHR   #^M<R2,R3>                                  ;
        00000000G    00             04  FB 0004E        CALLS   #4, TURN_WINDOW                             ;
                     50        08   A2  3C 00055        MOVZWL  8(WINDOW), R0                               ; 1168
                     50             30  C2 00059        SUBL2   #48, R0                                     ;
                     50             06  C6 0005C        DIVL2   #6, R0                                      ;
                     55        16   A2  3C 0005F        MOVZWL  22(WINDOW), FREE_POINTERS                   ;
55                   50             55  C3 00063        SUBL3   FREE_POINTERS, R0, FREE_POINTERS            ;
                     50        30   A2  3C 00067        MOVZWL  48(WINDOW), R0                              ; 1169
                     50        2C   A2  C0 0006B        ADDL2   44(WINDOW), R0                              ;
                     03             50  D1 0006F        CMPL    R0, #3                                      ;
                                    18  1A 00072        BGTRU   1$                                          ;
                                    55  D6 00074        INCL    FREE_POINTERS                               ; 1172
                     50        30   A2  3C 00076        MOVZWL  48(WINDOW), R0                              ; 1173
                     50        2C   A2  C0 0007A        ADDL2   44(WINDOW), R0                              ;
                     51        36   A2  3C 0007E        MOVZWL  54(WINDOW), R1                              ;
                     50             51  C0 00082        ADDL2   R1, R0                                      ;
                     03             50  D1 00085        CMPL    R0, #3                                      ;
                                    02  1A 00088        BGTRU   1$                                          ;
                                    55  D6 0008A        INCL    FREE_POINTERS                               ; 1174
                                    55  D5 0008C 1$:    TSTL    FREE_POINTERS                               ; 1176
                                    03  14 0008E        BGTR    2$                                          ;
                     55             01  D0 00090        MOVL    #1, FREE_POINTERS                           ;
                     50             67  D0 00093 2$:    MOVL    (R7), R0                                    ; 1182
```

```
                51      38  A0  9A 00096          MOVZBL  56(R0), R1                          ;
        51      51          0C  78 0009A          ASHL    #12, R1, R1                         ;
                50      44  A0  D0 0009E          MOVL    68(R0), R0                          ;
                51          50  D1 000A2          CMPL    R0, R1                              ;
                            03  1B 000A5          BLEQU   3$                                  ;
                50          51  D0 000A7          MOVL    R1, R0                              ;
                50      38  A4  C2 000AA 3$:      SUBL2   56(FCB), R0                         ; 1183
                51          67  D0 000AE          MOVL    (R7), R1                            ; 1184
                52      38  A1  9A 000B1          MOVZBL  56(R1), R2                          ;
                52          50  C0 000B5          ADDL2   R0, R2                              ;
                50      3C  A1  3C 000B8          MOVZWL  60(R1), R0                          ;
                50        6240  DE 000BC          MOVAL   (R2)[R0], FILES_TO_GO               ;
                            05  14 000C0          BGTR    4$                                  ; 1185
                      08D0  8F  BF 000C2          CHMU    #2256                               ; 1186
                            04  000C6             RET                                         ;
                51          67  D0 000C7 4$:      MOVL    (R7), R1                            ; 1196
                52      3C  A1  3C 000CA          MOVZWL  60(R1), R2                          ;
        51      40  A1          52  C7 000CE      DIVL3   R2, 64(R1), R1                      ;
                51              04  C6 000D3      DIVL2   #4, R1                              ;
                52              50  D0 000D6      MOVL    FILES_TO_GO, R2                     ;
                51              52  D1 000D9      CMPL    R2, R1                              ;
                                03  1B 000DC      BLEQU   5$                                  ;
                52              51  D0 000DE      MOVL    R1, R2                              ;
                50              52  D0 000E1 5$:  MOVL    R2, FILES_TO_GO                     ; 1195
                50              55  C6 000E4      DIVL2   FREE_POINTERS, R0                   ; 1198
        000003E8  8F            50  D1 000E7      CMPL    R0, #1000                           ;
                                05  1B 000EE      BLEQU   6$                                  ;
                50      03E8    8F  3C 000F0      MOVZWL  #1000, R0                           ;
            18  A6              50  D0 000F5 6$:  MOVL    BLOCKS_NEEDED, 24(FIB)              ; 1203
            16  A6      820B    8F  A8 000F9      BISW2   #33291, 22(FIB)                     ; 1211
                                53  DD 000FF      PUSHL   HEADER                              ; 1213
                                56  DD 00101      PUSHL   FIB                                 ;
            0000G   CF          02  FB 00103      CALLS   #2, EXTEND                          ;
        56  18  A6          1C  A6  C1 00108      ADDL3   28(FIB), 24(FIB), R6                ; 1215
                                56  D7 0010E      DECL    FILESIZE                            ;
    18  A3          56          10  9C 00110      ROTL    #16, FILESIZE, 24(HEADER)           ; 1216
                    50          67  D0 00115      MOVL    (R7), R0                            ; 1217
        09  53  A0              04  E0 00118      BBS     #4, 83(R0), 7$                      ;
                    50      01  A6  9E 0011D      MOVAB   1(R6), R0                           ; 1218
    1C  A3          50          10  9C 00121      ROTL    #16, R0, 28(HEADER)                 ;
                    28          63  91 00126 7$:  CMPB    (HEADER), #40                       ; 1224
                                05  1F 00129      BLSSU   8$                                  ;
            4C  A3          01  A6  9E 0012B      MOVAB   1(R6), 76(HEADER)                   ; 1226
                                53  DD 00130 8$:  PUSHL   HEADER                              ; 1231
            0000G   CF          01  FB 00132      CALLS   #1, CHECKSUM                        ;
            0000G   CF          00  FB 00137      CALLS   #0, WRITE_HEADER                    ; 1232
                    50          67  D0 0013C      MOVL    (R7), R0                            ; 1233
                            2C  A0  DD 0013F      PUSHL   44(R0)                              ;
                                53  DD 00142      PUSHL   HEADER                              ;
            0000G   CF          02  FB 00144      CALLS   #2, RESET_LBN                       ;
                                53  DD 00149      PUSHL   HEADER                              ; 1234
            0000G   CF          01  FB 0014B      CALLS   #1, WRITE_BLOCK                     ;
                                53  DD 00150      PUSHL   HEADER                              ; 1235
            0000G   CF          01  FB 00152      CALLS   #1, INVALIDATE                      ;
                                53  DD 00157      PUSHL   HEADER                              ; 1236
                                54  DD 00159      PUSHL   FCB                                 ;
            0000G   CF          02  FB 0015B      CALLS   #2, INIT_FCB2                       ;
```

```
                                18   AA  DD 00160        PUSHL   24(BASE)                                  ; 1238
                        0000G   CF      01  FB 00163    CALLS   #1, RELEASE_SERIAL_LOCK                   ;
                                18   AA  D4 00168       CLRL    24(BASE)                                  ; 1239
                        0000G   CF      00  FB 0016B    CALLS   #0, RESTORE_CONTEXT                       ; 1241
                                04   A8  D4 00170       CLRL    4(R8)                                     ; 1242
                                         04 00173       RET                                               ; 1244

; Routine Size:  372 bytes,    Routine Base:  $CODE$ + 0000


;  256          1245  1
;  257          1246  1 END
;  258          1247  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;  $CODE$                          372  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                               Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1         18619        49         0      1000          00:01.9



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:EXTIDX/OBJ=OBJ$:EXTIDX MSRC$:EXTIDX/UPDATE=(ENH$:EXTIDX)

; Size:          372 code + 0 data bytes
; Run Time:        00:20.0
; Elapsed Time:    00:38.7
; Lines/CPU Min:    3746
; Lexemes/CPU-Min: 45989
; Memory Used:  260 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTIDX
LIS

GTLCAT
LIS

GETLOC
LIS

EXTEND
LIS

EXTHDR
LIS

FILUTL
LIS

GETREQ
LIS

EXTCONTIG
LIS

ERASE
LIS

GETTIM
LIS

FIND
LIS

GETFIB
LIS

INIFC2
LIS

INIFCP
LIS

EXTFCB
LIS

FILESERV
LIS

FILESIZE
LIS

GETPTR
LIS